

Ruby master - Feature #12046

Allow attr_reader :foo? to define instance method foo? for accessing @foo

02/02/2016 04:24 AM - mrkn (Kenta Murata)

Status:	Rejected	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
Now we should write an accessor method of a boolean instance variable like		
<pre>def foo? @foo end</pre>		
But I want to write it by using attr_reader like:		
<pre>attr_reader :foo?</pre>		
Related issues:		
Related to Ruby master - Feature #11167: Allow an attr_ variant for query-met...		Rejected
Related to Ruby master - Feature #5781: Query attributes (attribute methods e...		Assigned
Related to Ruby master - Feature #15991: Allow questionmarks in variable names		Open
Is duplicate of Ruby master - Feature #10720: A proposal for something like: ...		Rejected

History

#1 - 02/02/2016 04:55 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #10720: A proposal for something like: attr_reader :foo? - with the trailing '?' question mark added

#2 - 02/07/2016 02:00 PM - shevegen (Robert A. Heiler)

Yay! Sometimes things just take a while. :)

#3 - 02/16/2016 06:53 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

as a rule, attr_reader x creates an instance variable @x, but we cannot have @x?. that's the reason, we reject attr_reader :foo?. If you have any concrete use-case for the new (more complex) behavior, please tell me.

Matz.

#4 - 10/29/2016 01:37 PM - snoop (Sven Schwyn)

If you have any concrete use-case for the new (more complex) behavior, please tell me.

This feature request has been rejected more than once, so I'm most likely doing this in vain. But the obvious use-case would be better readability. Right now, the way to do it is:

```
class Foobar  
  attr_accessor :color  
  attr_writer :transparent  
  
  def transparent?  
    @transparent  
  end  
end
```

Even though the transparent reader is trivial, it has to be explicitly added and thus is not part of the list of other trivial accessors such as color.

If trailing question marks are ignored when trivial writers are created, the code looks as follows:

```
class Foobar
  attr_accessor :color, :transparent?
end
```

The writer remains "transparent=", but it could reject any non-boolean values. The reader would be "transparent" with an alias "transparent?"

It's just a little magic to remove crust. And it wouldn't break any existing code because accessors and writers with trailing question marks raise "invalid attribute name" as of now.

#5 - 07/09/2019 12:08 AM - mame (Yusuke Endoh)

- Related to Feature #11167: Allow an attr_ variant for query-methods that end with a question mark '?' character, such as: `def foo? returning @foo` added

#6 - 07/09/2019 12:08 AM - mame (Yusuke Endoh)

- Related to Feature #5781: Query attributes (attribute methods ending in '?' mark) added

#7 - 08/28/2019 12:52 AM - mrkn (Kenta Murata)

- Related to Feature #15991: Allow questionmarks in variable names added